

# **SEMESTER TRAINING REPORT**

ON

## **MEDICOS**

*Submitted in partial fulfillment of requirements  
for the award of the degree*

*Bachelor of Technology*

*In*

**Computer Science and Engineering**

To

**Punjab Technical University, Jalandhar**

**SUBMITTED BY:**

**Name: Anurag**

**Roll no.: 1801994**

**Semester: 8th**

**Batch: 2018-2022**

**Under the guidance of  
Mr. Sachin Bhardwaj**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**Chandigarh Engineering College –Landran**

**Mohali, Punjab - 140307**

## **CERTIFICATE**

This is to certify that Mr. Anurag has partially completed / completed / not completed the Semester Training during the period from January to July in our Organization / Industry as a Partial Fulfillment of Degree of Bachelor of Technology in Computer Science & Engineering.

**(Signature of Project supervisor )**

**Date:** \_\_\_\_\_

## DECLARATION

I hereby declare that the Project Report entitled MEDICOS is an authentic record of my own work as requirements of 8<sup>th</sup> sem academic during the period from January to July for the award of degree of B. Tech. ( Computer Science & Engineering , **Chandigarh Engineering College, Landran** under the guidance of Mr. Sachin Bhardwaj.

**Anurag**  
**1801994**

**Date:** \_\_\_\_\_

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

## Signatures

**Examined by:**

1.

2.

3.

4.

**Head of Department  
(Signature and Seal)**

## ACKNOWLEDGMENT

I take this opportunity to express my sincere gratitude to the Director- Principal Dr. Rajdeep Singh Chandigarh Engineering College, Landran for providing this opportunity to carry out the present work.

I am highly grateful to the Dr. Sukhpreet Kaur HOD CSE, Chandigarh Engineering College, Landran (Mohali). I would like to express my gratitude to other faculty members of Computer Science & Engineering department of CEC, Landran for providing academic inputs, guidance & Encouragement throughout the training period. The help rendered by **Mr. Rajeev Sharma**, Supervisor for Experimentation is greatly acknowledged. Finally, I express my indebtedness to all who have directly or indirectly contributed to the successful completion of my software training.

## **About Company**

AlmaBetter is a global technology company headquartered in Bangalore India. Our comprehensive portfolio of digital products and services is backed by our unrelenting focus on problem-solving and unique approach to engineering. AlmaBetter offer agile and open source solutions that help our partner companies embrace digital, meet evolving industry needs and power business transformation at an optimized cost. We develop highly scalable and performance driven applications in Web, Mobile, Blockchain, Big Data and Analytics for Bay Area Startups and Fortune 500 companies. Our focused, enthusiastic and expert team of professionals across design, development, project management, quality assurance and sales and marketing work as a unit to help clients meet their business.

## TABLE OF CONTENTS

<i>Certificate</i>	<i>ii</i>
<i>Declaration</i>	<i>iii</i>
<i>Acknowledgment</i>	<i>iv</i>
<i>About Company</i>	<i>v</i>
<i>List of Figures</i>	<i>vii</i>

## CHAPTERS

1	Introduction to Project Feasibility Analysis	08
2	Software Requirement Specification	10
3	Introduction and detail of software used	12
4	System Design	18
5	System Implementation And Testing	20
6	Implementation	23
7	Testing	26
8	Screenshots	27
9	Conclusions and Future Scope	38
10	References/ Bibliograph	40

## LIST OF FIGURES

FIGURE NO.	DESCRIPTION	PAGE NO.
1.	ERD	17
2.	Context Analysis diagram	18
3.	DFD: Level 1	18
4.	DFD: Level 2	19
5.	Login Display	19

# Chapter 1

## Introduction

Blockchain is defined as a ledger system that aids in the management and storage of data in time-stamped blocks that operate mostly decentralized across any computing networks and are linked via encryption. Bitcoin [Nakamoto, 2009], the first electronic payment system to properly exploit the power of blockchain technology, was introduced in the fintech business. Since then, blockchain technology has permeated several industries of information technology (IT). Blockchain technology has the potential to transform several industries, including finance, governance, health care, and supply chain. Its anticipated benefits include lower transaction costs and complexity, more security, greater transparency, and regulation.

Healthcare is considered as one of the blockchain technology's application areas. However, technological adoption in the healthcare industry is slow. Healthcare is characterized as a conventional sector that is difficult to measure due to changing facts and resistant to innovation techniques. Healthcare issues (such as privacy, quality of care, and information security) have received a lot of attention in recent years. Care coordination between patient and health care provider is becoming more difficult as the prevalence of chronic diseases in an aging and rising population rises. In many cases, the technology available in health care is insufficient to record all types of service provided. This is mostly due to the usage of outdated technologies to convey information between parties. To collect and exchange medical data, health care practitioners continue to rely on outdated systems and paper-based medical records.

Health care providers continue to invest significant resources in processing medical claims and administrative data, despite the fact that most of this may be eliminated



utilising technology such as BlockChain. Blockchain technology is rapidly being recognised as a means for addressing existing challenges with information distribution.

In this paper, We propose Medicos, an health data application based on blockchain technology, where health data are stored with different levels of access controls and privacy policies to give the best benefits. In Medicos admins (hospital management) or IoT devices can regularly update the users' health data. Doctors and AI health analyzers can anywhere and anytime diagnose these data and this data will build up all the historical data of that particular user and will be connected by a unique identification number (Aadhar no.).

# Chapter 2

## Software Requirement Specification

### **Summary**

Medicos, an health data application based on blockchain technology, where health data are stored with different levels of access controls and privacy policies to give the best benefits. In Medicos admins (hospital management) or IoT devices can regularly update the users' health data. Doctors and AI health analyzers can anywhere and anytime diagnose these data and this data will build up all the historical data of that particular user and will be connected by a unique identification number (Aadhar no.) to every patient. The data will continuously keep on growing and blockchain is incapable of storing huge amounts of data as the resource requirements will become too high and complexity will increase to maintain, search and verify. The optimal solution at this point will be to use the InterPlanetary File System (IPFS), which is a content-addressable, distributed file system to store data. The data is distributed over different nodes in the network as there is no central server. Even if some nodes are disconnected still the data is accessible from other running nodes and it can distribute large amounts of data without duplication. Every file uploaded to IPFS has a unique hash string through which the file can be retrieved. In the Medicos application, most of the data is stored in the IPFS and that particular hash of that file is stored in the blockchain and helps to verify data integrity and map the data in the ipfs storage. The health data has to be accessed by the doctors and health analyzers so access control is added to the application so only specific groups of people can access the sensitive health data.

### **Departments to be involved**

1. Admin
2. Doctors

### **Target Audiences**

Healthcare Sector

### **Project Motivation**

1. Some medical records are lost, mutated & tampered

2. Some hospitals show wrong results and charge a lot of money to treat it.
3. Health records are not connected to a unique identification no. across the whole country and each hospital is maintaining a different set of records that needs to be unified.
4. The history of a patients' data when trying to retrieve at a later stage takes a lot of time and not everything shows up.

### **Project Objective**

1. Medical records cannot be mutated, deleted, or tampered with anymore as the database is getting decentralized with the help of blockchain technology.
2. Nobody at a later stage can show different results as once written on blockchain nothing can be changed.
3. Each health record is connected to a user's unique identification number or Aadhar no. and we can get a unified & decentralized database all across the country.
4. Easily all the health records of a patient can be retrieved by searching by his aadhar no.

# Chapter 3

## Introduction and Details of Software Used

### **Introduction**

With the dramatic increase in the Internet of Things (IoT), remote monitoring of health data and automation of health data to achieve intelligent healthcare solutions has received great attention recently. Due to the limited computing power and storage capacity of IoT devices, users' health data are generally stored in centralized third-party applications, such as the hospital database or cloud databases, and make users lose control of their health data, which can easily result in privacy leakage and single-point bottleneck of getting the data erased, mutated or deleted. In this paper, I propose Medicos, a large-scale health data application based on blockchain technology, where health data are stored with different levels of access controls and privacy policies to give the best benefits. Specifically, admins can effectively revoke or add authorized doctors. Furthermore, by introducing Medicos, both IoT data and doctor diagnosis cannot be deleted or tampered with so as to avoid medical disputes and having no change in the history of medical treatments. Theoretical analysis and experimental results show that the proposed Medicos is applicable for smart healthcare systems. This system is ready to be highly scalable and can be massively adopted and what is proposed in this paper is just a prototype of the same.

## **TECHNOLOGY USED**

The main technologies implemented in this project are: Blockchain, Inter-Planetary File System, Ethereum, Smart Contracts, Ethereum Virtual Machine, Solidity, React Javascript, Ganache, Metamask, Github, Consensus Algorithm, and Hashing Algorithm. A brief description of all these technologies follows:

**Blockchain:** It is a distributed ledger-based technology that uses consensus-based decisions to come to a single point of truth. It involves three main technologies which are private key cryptography, peer-2-peer network, and Blockchain protocol. The data once entered becomes immutable and hackproof. Blocks in blockchain contain groups of valid transactions that are encoded into a Merkle tree structure and hashed at each step of the merkle tree generation. Each block contains the hash of the parent block in the blockchain, linking the two together so that they cannot be tampered. The linked blocks form a chain. This repetitive process confirms the integrity of the parent or previous block, all the way back to the genesis block. Each block also has a timestamp and a nonce associated with it.

**Inter-Planetary File System:** The Inter-Planetary File System[IPFS] protocol is a peer-2-peer network for sharing and storing data in a distributed system of files. For uniquely identifying each file it uses content addressing in a global namespace in which all computing nodes are connected in a decentralized manner. It uses content-based addressing and Merkle Directed Acyclic Graph data structure.

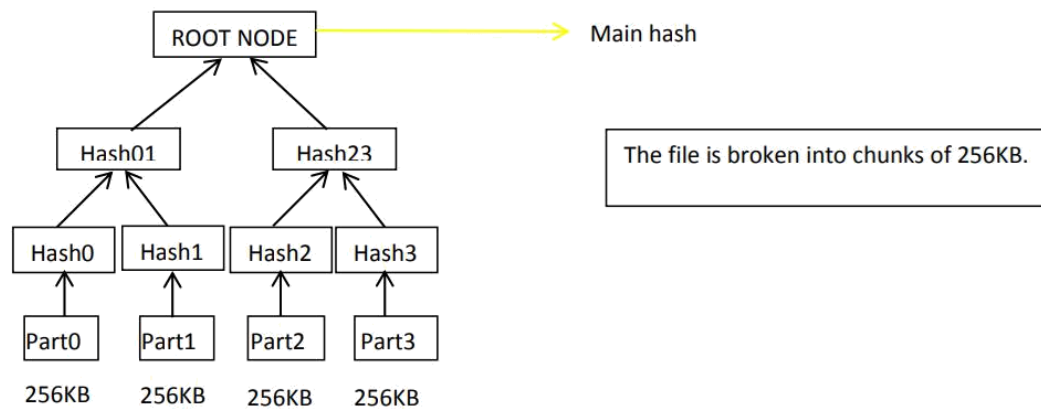


Fig 5: Storing strategy in IPFS via Merkle-DAG

A Merkle-DAG is a Directed Acyclic Graph structure where each node has an identifier and this is the result of hashing the contents of the node. So to get the hash of the any node the children nodes are hashed together and the new hash that comes is the hash of the parent node. The hashing is done using a cryptographic hash function like SHA256.

**Ethereum:** Ethereum is a platform based on Blockchain technology that enables anyone to build and deploy smart contracts and connected decentralized applications. Ethereum’s coding language solidity helps write smart contracts. Its native currency is Eth. It was founded by Vitalik Buterin.

**Dapplication:** A decentralized application is a computer application that runs on a distributed computing system. They have distributed ledger [DLT] based technology. It has a web-front in any javascript libraries, smart contract running on blockchain as backend and web3, or some kind of RPC pipelines to connect both of them together as a middleware.

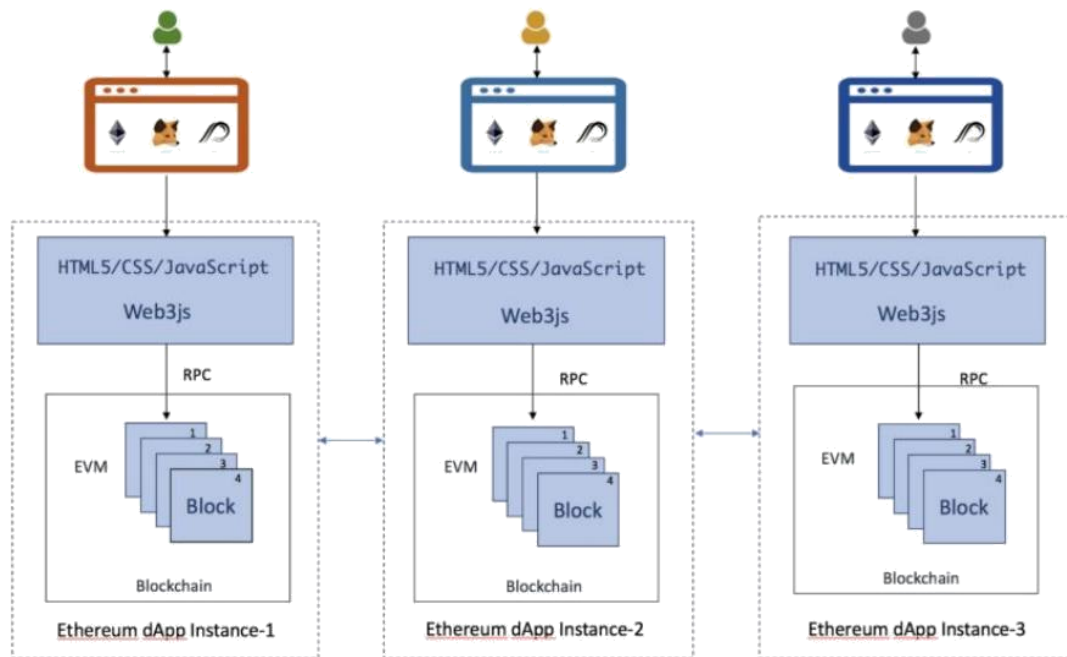


Fig 6: Ethereum Dapp instance

**Solidity:** Solidity is a contract oriented language. It is used mostly for Ethereum Virtual Machine. It is statically typed language so you have define types for each data and functions, supporting inheritance, libraries and complex user defined types. Almost all the applications of solidity are for writing smart contracts.

**GitHub:** It is a version control software used for versioning and hosting of software, codebases and applications. It offers the distributed version controlling and project code management functionality. In this project github is purely used for version control and first step in CI/CD pipeline.

**React.js:** React is a front-end open-source JavaScript library for building user experience and interfaces based on small components. In this project it is extensive used for the creation of front end and UI.

**Ganache:** Ganache is a copy of Ethereum Blockchain which can be used as a mock for personal use and is used to test and deploy smart contracts , develop applications, run tests and perform other functionalities free of cost. It is used in this project as the blockchain running locally.

**Metamask:** MetaMask is a cryptocurrency wallet for interacting with the Ethereum blockchain. It enables users to access their Ethereum wallet through a mobile app or browser extension, which can then be used to interact with decentralized applications. In this system it acts as a gateway to the blockchain world and to carry on all the transactions.

**Consensus Algorithm:** A voting mechanism where all the nodes connected to a network vote on the validity of a block only then the block is confirmed over the blockchain. The consensus algorithms that can be used for this project are Practical Byzantine Fault Tolerance, Proof of Work and Proof of stake.

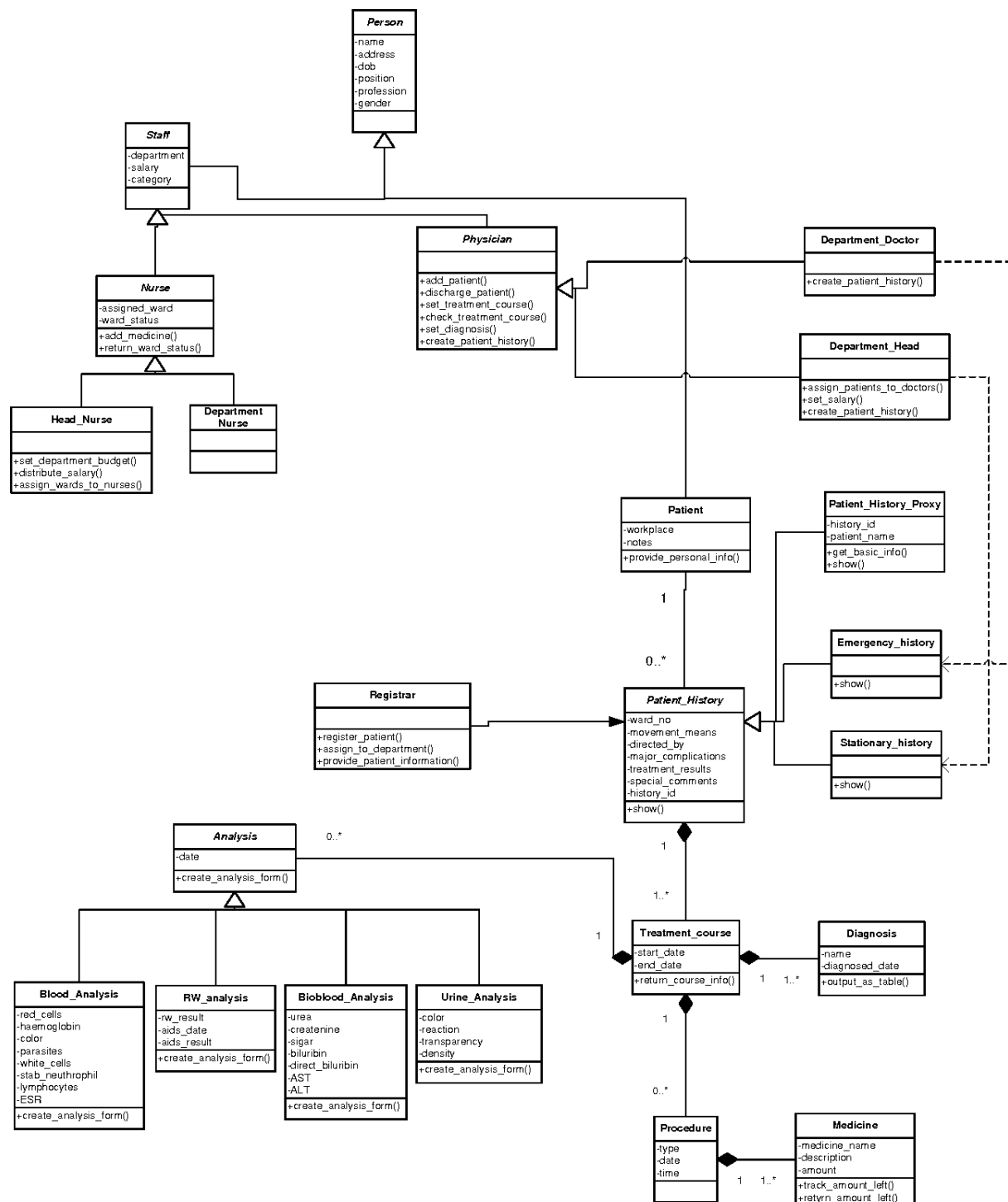
**Hashing Algorithm:** It plays a crucial role in the blockchain process and also in the integrity of the transaction and confidentiality of data. It transforms and maps an arbitrary length of input arbitrary data value to a unique fixed-length value. The algorithm should be one-way and collision-free. Some majorly used hashing functions are SHA-256 and Keccak.

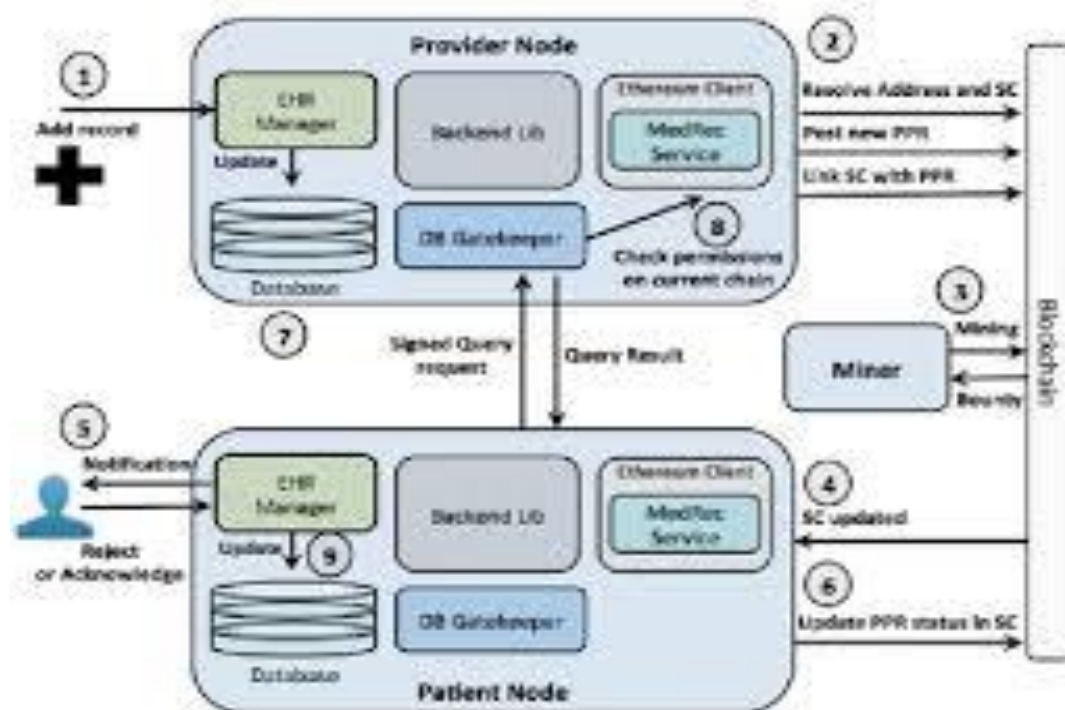
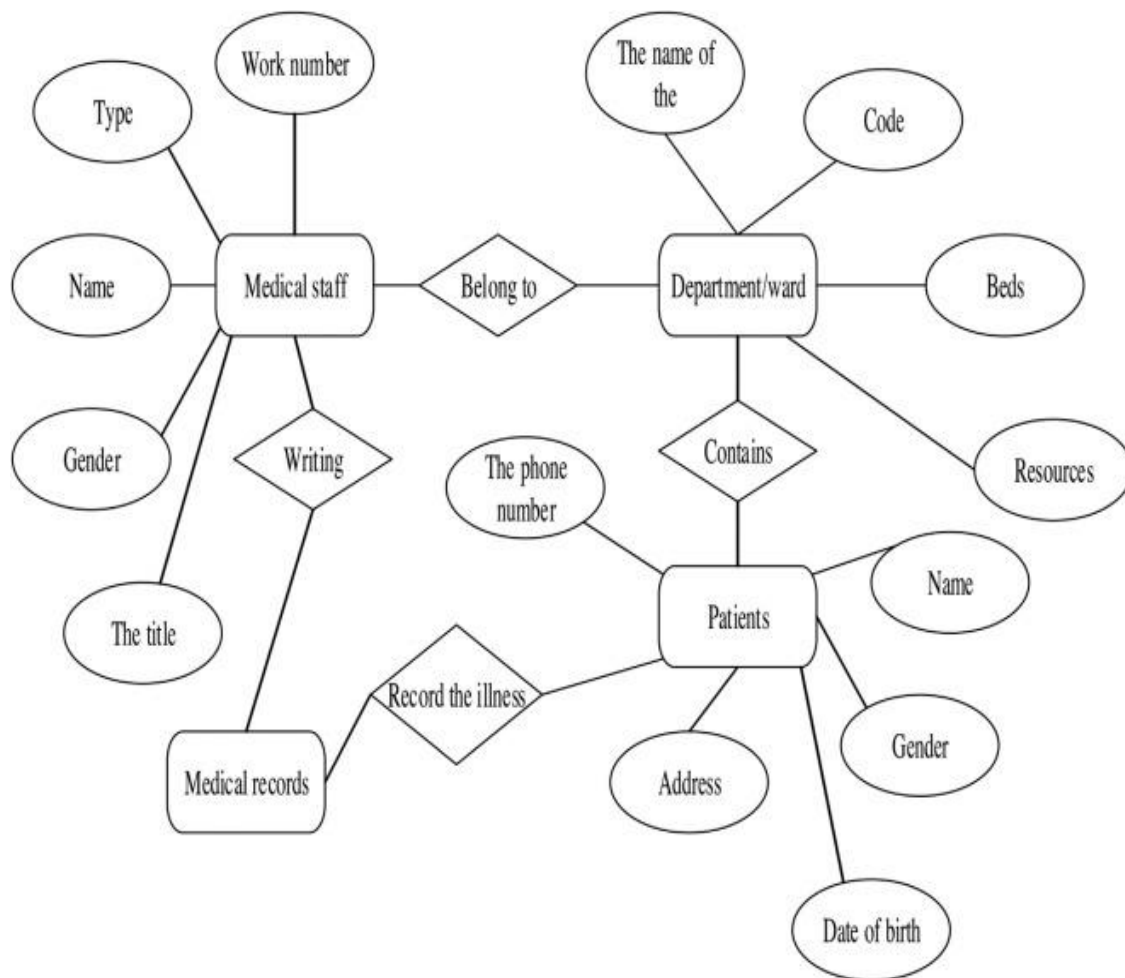
The implementation is done in the form of a small working prototype application. Here is a link for the workflow of each type of user and each functionality that I created for a better understanding of how the application works.

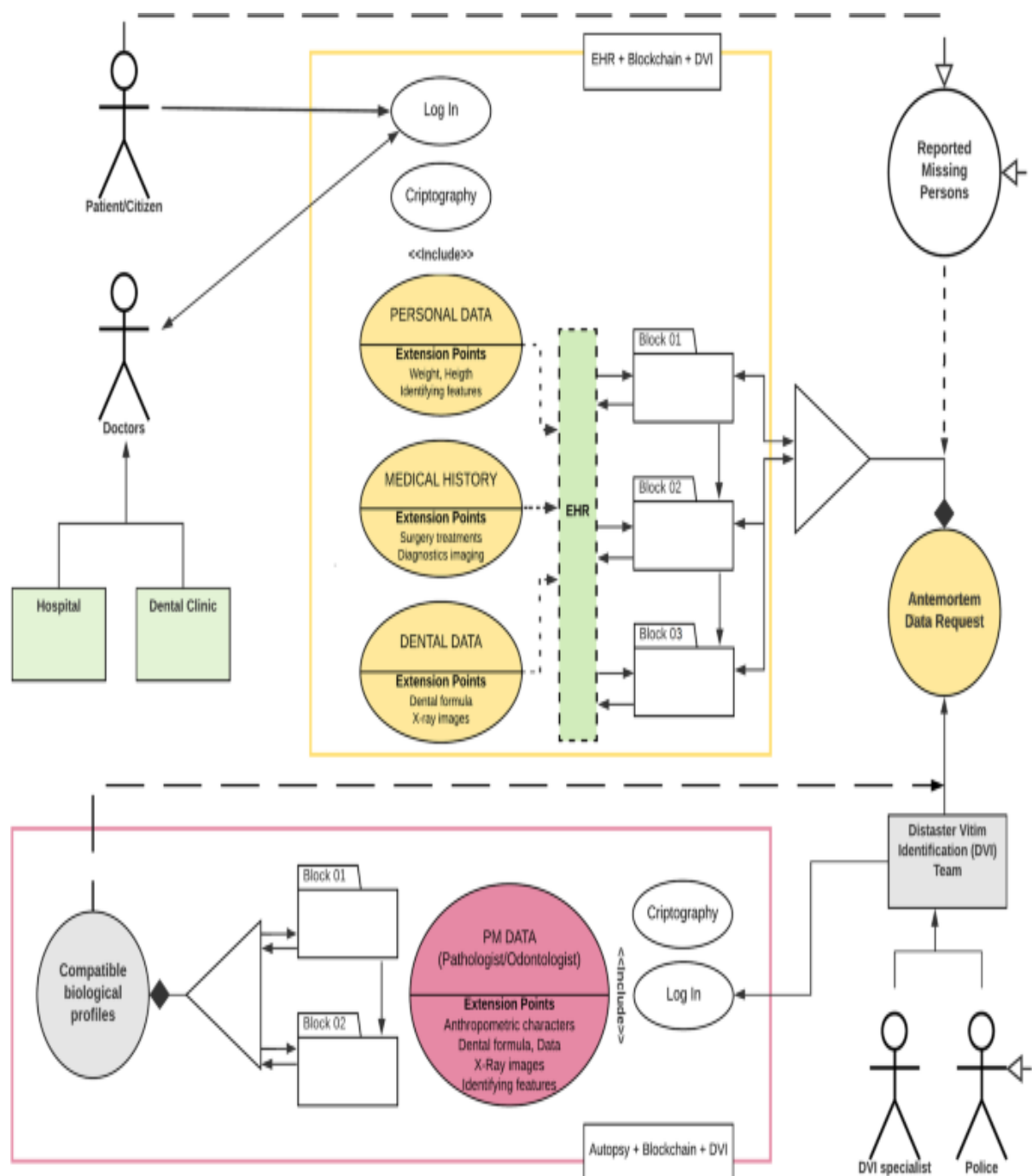


# System Design

**ERD**







# Chapter 5

## System Implementation and Testing

1. **Patient:** A person who is having to go to the hospital. Should have a unique identification no. such as aadhar number which is already provided by the government to all its citizens.
2. **Doctor:** A doctor who can treat patients and is associated with a hospital or is isolatedly having his own clinic.
3. **Hospital Staff:** A hospital staff helps in the process of registering patients and doctors on the application. Helps in the retrieval of health data of patients and doctors also add treatment for patients.

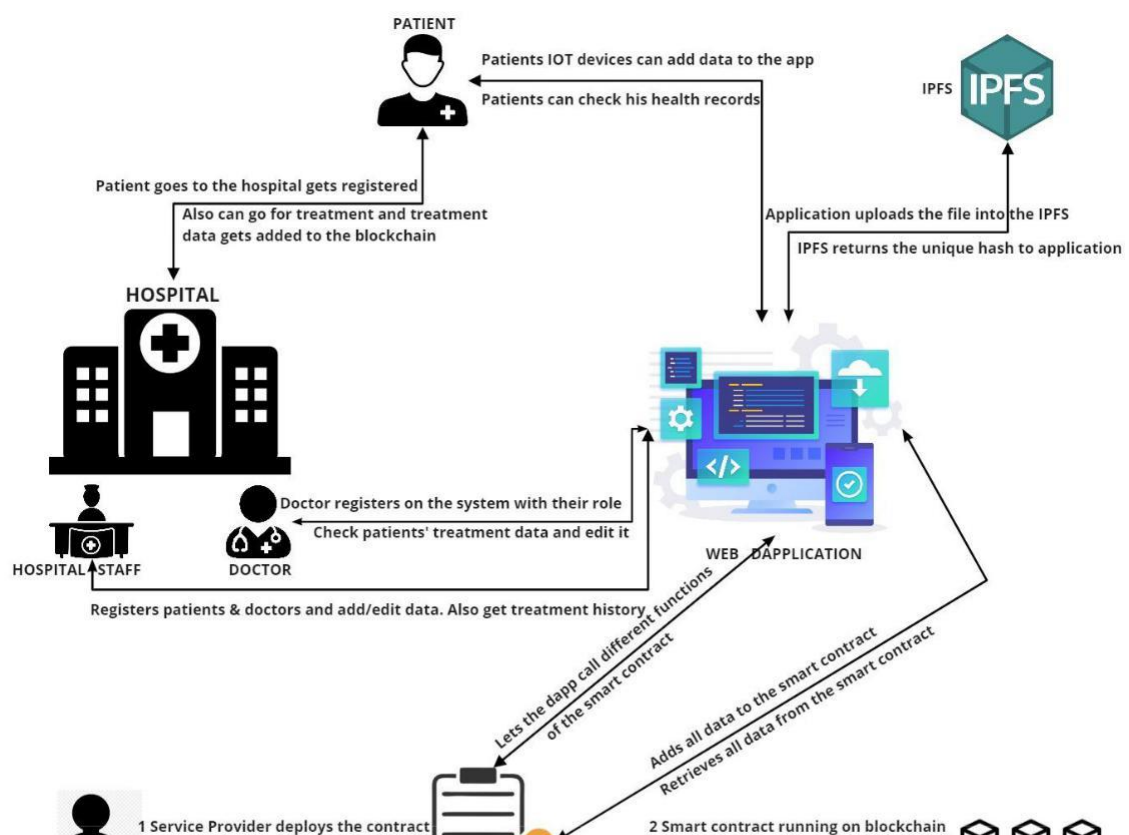


Fig 4 : Proposed blockchain architecture

Figure 4 shows the detailed model of the proposed system. Currently I'm using a local blockchain known as ganache which works exactly as Ethereum blockchain so that all transactions run very fast it is also running on rinkeby & ropsten network of ethereum to

which everybody involved in the system can get connected but is not compulsory and the only difference it has from the ethereum mainnet is that it only requires fake money and not the real one. I developed a smart contract written in Solidity language to manage all these activities. This smart contract is running on the "Code is law" protocol i.e. the functionality under all circumstances will follow whatever is written in the code. This contract is deployed on ganache blockchain, rinkeby and ropsten testnets of Ethereum network also. I created a web dapplication using React.js [A javascript library for building user interfaces] for the frontend creation for all the user interfaces and functionality that can be leveraged for this application, Web3.js [a library that allows you to interact with local or remote Ethereum node using HTTP, IPC, or Websocket] connected with blockchain backend using Metamask[an extension for accessing Ethereum enabled distributed applications, or "Dapps" in your browser] extension that allows you to cover for the gas fees and transaction cost. It is used to be an interface for all the involved parties to access the blockchain network and carry out various activities in the Medicos system. The system strictly follows the decentralized aspects of a private blockchain with a consensus mechanism of entities for every transaction. The detailed steps involved in our proposed Blockchain system are depicted by fig. 4 as follows:

**Mining Continuously:** The process of block production in blockchain is known as mining and is carried out by miners who validate all transactions and check the legitimacy of all functions getting called in the smart contract. In our platform, the mining is continuous in the blockchain and smart contract written in solidity language accomplishes code is law where the functions runs exactly as it is coded in the contract. The service provider can deploy the contract[once and for all] and start mining. The

access control is not at the mining level rather it is in the contract layer. So the mining is continuous but the transactions can only be done by a particular agent and the smart contract's validates all transactions before sending them to the blockchain.

**Users Registry:** This step is to collect all the information about the involved parties, which can be achieved by the registration of all the parties such as hospital staff, doctors, and patients into the platform. The registration can be carried out with a government identity proof(Aadhar number)of the parties and their registration details get stored in Blockchain. While registering the patients all health data is added to the blockchain and ipfs [and the link is sent into the blockchain] which lets the blockchain store only hashes and ipfs storing all the bigger files which might take a lot of storage.

**Adding Treatment:** Whenever a patient comes for any type of treatment at the hospital, the hospital staff adds a new treatment [the details of all the treatment] and it gets connected to the patient's unique no. [aadhar no.] and the set of doctors doing the treatment also gets connected to that treatment. So everything in this system is strongly interleaved with each other and can be retrieved very easily only by concerned parties.

**Changing Treatment Stage:** Treatment status can only be advanced to next steps and done by only hospital staff and doctors and it tracks the history of the treatment as the history cannot be mutated or deleted in blockchain. We solve the purpose of losing or getting records manipulated. At each stage new doctors might get added, new reports and prescriptions also get added and can be tracked anytime. The system is made robust enough to handle all these.

# Chapter 6

## Implementation

- 1 we have the member list and only the hospital staff can access it. They can see the details of all the hospital staff as in for now only the wallet address and aadhar no. but later the details can be increased also. In addition to this even the doctors' details namely aadhar number, wallet address, and role can be seen
- 2 Next page is exclusively for patients' registration into the system for the first time and later on it can be updated and each time there is a new treatment. In this registration, the patient has to give his aadhar no., weight (in grams), height (in cms), gender, blood group, date of birth in epoch time, and location. After the hospital staff or the doctor enters this data it will go into the blockchain as a transaction. There is a validation for kind of extreme values for each field. If the value deviates from general values by a lot its going to throw an error statement. For example the Height cannot be more than 300 cms so we have the error handler on it which throws an error statement anytime somebody enters values exceeding the limit.
- 3 Next page, the hospital staff, doctor or the patient can retrieve the patient's data by putting his aadhar no. in the search box and clicking on "Get Data" and all the details will show in the below section. The patient's aadhar no., height, weight, bloodtype, gender and location is shown. In this section there is "Treatments Undergone" which will show all the treatments Id this patient has gone through.

- 4 Next page where hospital staff or doctors can add treatments. They should start by adding the patient's aadhar when he arrives at the hospital for some treatment. Once the add treatment is clicked a treatment is generated in the blockchain which has the treatment Id, the patient's aadhar no. and the logged in user's aadhar no. (hospital staff or doctor whoever did the transaction). Whenever this treatment is assigned to a doctor the "add doctor" form should be used in which the treatment id and doctor's aadhar no. should be added who is going to do the treatment and this data will go to the blockchain and also an event will be emitted in the blockchain to make this doctor addition immutable. Next during the whole process of this treatment any prescription or reports can be added just by adding the treatment Id and uploading the file for the same (any format) and then click on add. This file is first uploaded to Ipfs from where ipfs gives back a hash through which the file can be accessed. This hash is then sent to the blockchain to save up on space by keeping just hash rather than whole file.
- 5 Next page shows all the treatments in the system. It shows just basic data namely the treatment Id, Patient's aadhar and the admin's aadhar who added this treatment. There is a search bar in which when a patient's aadhar no. is put and searched for it will return only that particular patient's treatments. Every single treatment when clicked routes to a Treatment details page which is shown in the next image.
- 6 Next is the Treatment History/Details page it firstly shows the treatment Id, Patient's aadhar and the admin's aadhar who added this treatment. Then in the events section it gives all the history of this treatment since when it was added and timestamp of each of them. In this particular example it shows when the treatment was added. It shows that a doctor was added to the treatment and his aadhar no. following a prescription and report was added and shows the time of it. The picture is a small version of the actual picture which can be viewed by clicking on this image and this image is getting retrieved from the ipfs by taking the hash from the blockchain. Also the hash is shown in the same card.



- 7 Next is Ganache application where I'm currently running a local blockchain and it functions exactly the same as the ethereum blockchain just with fake money and I'm using it for testing the application, seeing logs, having accounts and funds in them.
- 8 Next is from my github repository which I'm using as a version control system and keeps my codes safe from getting deleted or modified.

Here is the link to my repository: <https://github.com/anuragsaini2472002/Medicos>

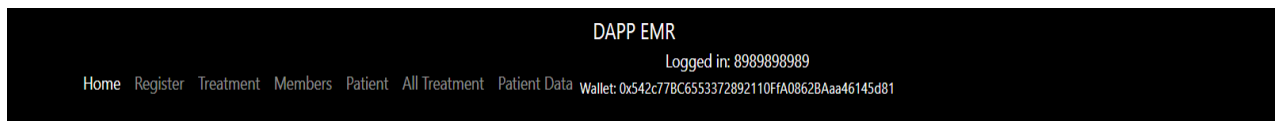
# Chapter 7

## Testing

- 1 Starting with Homepage. We have navbar with some more features.
  - 2 Registration page where the admin can signup/login/logout. All validation working properly also the wallet connect with Metamask.
  - 3 Member list can access it only hospital staff only the wallet address and unique aadhar number.
  - 4 In Patient registration page, firstly patient give his aadhar number, weight, height, gender, blood group, date of birth and location.
  - 5 In Patient detail page, doctor or the patient can retrieve the data putting his aadhar number and then clicking on “Get Data” button.
  - 6 In add treatment and reports page, hospital staff or doctors can add treatments. Also uploaded files in IPFS system.
  - 7 In Treatment details page, this page show all the treatment details.
  - 8 In Treatment history page, it shows when the treatment was added.
  - 9 In Ganache, transaction successfully mined also new block created with their block number.
-

# Chapter 8


## Screenshots



Welcome to Medicos Application

# SignUp Page

Admin



Aadhar Number:

8989898989


Role:

Hospital Staff

SIGN UP

Log InLog Out

Doctor



Wallet Address:

0xc22c6837dc526833D098b8e1247D966815238186

Aadhar Number:

5656565656

Speciality:

Cardiologist

Location:

Hyderabad

SIGN UP

Log InLog Out

## All Members

### Admins



Admin Aadhar:  
8989898989

Account:  
0xC28D1CbA696ACd6b085E  
9eDF41b433e117198333

Role: Hospital Staff 1

### Doctors



Doctor Aadhar:  
5656565656

Account:  
0xC22c6837dc526831D098b  
8e1247D966815238186

Role: Cardiologist

## Add Patient

Patient Aadhar	<input type="text" value="1212121212"/>
Weight	<input type="text" value="53000"/>
Height	<input type="text" value="168"/>
Gender	<input type="text" value="Male"/>
Blood Type	<input type="text" value="O"/>
Date of Birth	<input type="text" value="987558411"/>
Location	<input type="text" value="Mumbai"/>

X

DAPP EMR

Logged in: 8989898989

me Register Treatment Members Patient All Treatment Patient Data Wallet: 0xC2fD1CbA696ACd6b885E9eDF41b433e117198333

Patient Details

Patient Aadhar

1212121212

Get Data

Patient Details



Patient Aadhar : 1212121212

Height : 168

Weight : 53000

Bloodtype : O

Gender : Male

Location : Hyderabad

Treatments Undergone : 1,

## Add Treatment

Patient Aadhar

1212121212

Add Treatment

0

## Add Doctor

Treatment Id

1

Doctor Aadhar

5656565656

Send Treatment

## Add Prescription and Report

Treatment Id

1

Prescription Upload

Choose File

images2.jpg

Add

Treatment Id

1

Report Upload

Choose File

No file chosen

Add



## Patient Details

Patient Aadhar

1212121212

Get Data

## All Treatment



[Treatment ID : 1](#)

[Patient Aadhar : 1212121212](#)

[Admin Aadhar : 8989898989](#)

Treatment History



Treatment ID	/ 1
Patient Aadhar	/ 1212121212
Admin Aadhar	/ 8909090909

Events

Event: treatAdded  
Time: March 27th, 2022 at 10:45 PM

Event: doctorAddedTreat  
Doctor: 5656565656  
Time: March 27th, 2022 at 10:46 PM



Event: PrescriptionAddedTreat

Prescription:  
Qms2QINRuvVWELkzwWWt98RFwqGz0h  
QyP19h4omQ9DydLj

Time: March 27th, 2022 at 10:49 PM



Event: ReportAddedTreat

Report:  
QmsAQWwVUbWK3qtCn4kVu4p3Uo2S  
GwUpvbmfsQA04g-

Time: March 27th, 2022 at 10:50 PM

Ganache

—

□

×

ACCOUNTS

BLOCKS

TRANSACTIONS

CONTRACTS

EVENTS

LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK  
3

GAS PRICE  
2000000000

GAS LIMIT  
6721975

HARDFORK  
MUIRGLACIER

NETWORK ID  
5777

RPC SERVER  
HTTP://127.0.0.1:7545

MINING STATUS  
AUTOMINING

WORKSPACE  
NEW

SWITCH

MNEMONIC ⓘ  
dream ring excite bubble joke retreat flash scale field ginger season ocean

HD PATH  
m/44'/60'/0'/0'/account\_index

ADDRESS 0x542c77BC6553372892110FfA0862BAaa46145d81	BALANCE 100.00 ETH	TX COUNT 3	INDEX 0	
ADDRESS 0x10633d6c3D353846f1e495D0a0FDf6276DC88f05	BALANCE 100.00 ETH	TX COUNT 0	INDEX 1	
ADDRESS 0x46b00D70915f3b9CBA0c8e46A9DF69D5d5BB1CfC	BALANCE 100.00 ETH	TX COUNT 0	INDEX 2	
ADDRESS 0x656086526235997B4c4E372AACf563d71a4417f4	BALANCE 100.00 ETH	TX COUNT 0	INDEX 3	
ADDRESS 0xF66d493b2b2a014e1BB7c0960981379B6358Be18	BALANCE 100.00 ETH	TX COUNT 0	INDEX 4	
ADDRESS 0xEA7D5C499dd107C12e4D69d43c1933f67c0c0b5e	BALANCE 100.00 ETH	TX COUNT 0	INDEX 5	
ADDRESS 0x47E9754deEAB31532Da88B8cFbE85123050F680a	BALANCE 100.00 ETH	TX COUNT 0	INDEX 6	



Localhost 7545



Connected

Account 1

0x542...5d81



99.9987 ETH



Buy



Send



Swap

Assets

Activity



Send

Jun 6 • To: 0xf2f...836f

-0 ETH

-0 ETH

Backup your Secret Recovery Phrase to keep your wallet and funds secure.



Backup now

Search or jump to...

Pull requests

Issues

Marketplace

Explore

anuragsaini2472002 / Medicos

Private

Unwatch

Fork

Star

Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

Code

About

anuragsaini2472002 Update README.md

✓ builds 2 days ago

3 commits

contracts	Add files via upload	2 days ago
public	Add files via upload	2 days ago
src	Add files via upload	2 days ago
README.md	Update README.md	2 days ago
package-lock.json	Add files via upload	2 days ago
package.json	Add files via upload	2 days ago
truffle-config.js	Add files via upload	2 days ago

README.md

Anurag - BTech Final Project

Medicos, an health data application based on blockchain technology, where health data are stored with different levels of access controls and privacy policies to give the best benefits. In Medicos admins (hospital management) or IoT devices can regularly update the users' health data. Doctors and AI health analyzers can anywhere and anytime diagnose these data and this data will build up all the historical data of that particular user and will be connected by a unique identification number (Aadhar no.) to every patient. The data will continuously keep on growing and blockchain is incapable of storing huge amounts of data as the resource requirements will become too high and complexity will increase to maintain, search and verify. The optimal solution at this point will be to use the InterPlanetary File System (IPFS), which is a content-addressable, distributed file system to store data. The data is distributed over different nodes in the network as there is no central server. Even if some nodes are disconnected still the data is accessible from other running nodes and it can distribute large amounts of data without duplication. Every file uploaded to IPFS has a unique hash string through which the file can be retrieved. In the Medicos application, most of the data is stored in the IPFS and that particular hash of that file is stored in the blockchain and helps to verify data integrity and map the data in the ipfs storage. The health data has to be accessed by the doctors and health analyzers so access control is added to the application so only specific groups of people can access the sensitive health data.

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Environments

github-pages Active

Languages

JavaScript 77.9%

CSS 10.1%

Solidity 9.3%

HTML 2.3%

## Chapter 9

# Result Discussion, Conclusion & Future Work

### **Result Discussion**

The application is already able to achieve all the objectives that were mentioned before making this proposal. The Medical records cannot be mutated, deleted, or tampered with anymore as the database is decentralized with the help of blockchain. Nobody at a later stage can show different results as once written on blockchain nothing can be changed but later on new reports can be added, as you might have seen in the application every operation is having a timestamp. Each health record is connected to a user's unique identification number or Aadhar no. and we can get a unified & decentralized database all across the country. Easily all the health records of a patient can be retrieved by searching by his aadhar no. and is way faster and more reliable for patients' moving from place to place. A lot of reliability as well as robustness is added to the application. Its rather simple, functional User interface makes it easy for all the users to like and use it.

---

## **Conclusion**

I can conclude that this application after using the latest and the top technologies has a lot of potential and a large-scale implementation of this application is worth the efforts behind it. Also, the system once in production will be very reliable, efficient, and user-friendly. It drives today's medical records applications to all new levels and this application can have many more integrations so that other applications can be easily integrated into this application and they will become decentralized very easily. Decentralization is the new future and making applications decentralized is the first step into it. Blockchain as an emerging technology will take everything into the decentralized world.

## **Future Work**

A lot can be improved in the user interface (UI) and user experience (UX) to make it more user-friendly. A lot of applications (currently used in Healthcare) can be integrated or some major features can be added to make this application an all-in-one application for all healthcare-related tasks. The IoT part of the applications is also what will be incorporated next to take the application to a whole new dimension. New extensions will be added to the application to make it interoperable for diverse medical usage.

---

# Chapter 7

## Bibliography/References

- [1] Nakamoto, Satoshi “Bitcoin: A peer-to-peer electronic cash system”, Decentralized Business Review, Pages 21260, Year 2008
  - [2] K. Ito, K. Tago, and Q. Jin, “i-blockchain: A blockchain-empowered individual-centric framework for privacy-preserved use of personal health data,” in 2018 9th International Conference on Information Technology in Medicine and Education (ITME). IEEE, 2018, pp. 829–833.
  - [3] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Blockchain for secure ehrs sharing of mobile cloud-based e-health systems,” IEEE access, vol. 7, pp. 66 792–66 806, 2019.
  - [4] K. Fan, S. Wang, Y. Ren, H. Li, and Y. Yang, “Medblock: Efficient and secure medical data sharing via blockchain,” Journal of medical systems, vol. 42, no. 8, pp. 1–11, 2018.
  - [5] G. S. Aujla and A. Jindal, “A decoupled blockchain approach for edge-envisioned iot-based healthcare monitoring,” IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 491–499, 2020.
  - [6] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, “A patient agent to manage blockchains for remote patient monitoring,” Stud Health Technol Inform, vol. 254, pp. 105–115, 2018.
  - [7] A. Kumar, R. Krishnamurthi, A. Nayyar, K. Sharma, V. Grover, and E. Hossain, “A novel smart healthcare design, simulation, and implementation using healthcare 4.0 processes,” IEEE Access, vol. 8, pp. 118 433–118 471, 2020.
  - [8] X. Zheng, R. R. Mukkamala, R. Vatrappu, and J. Ordieres-Mere, “Blockchain-based personal health data sharing system using cloud storage,” in 2018 IEEE 20th international conference on e-health networking, applications and services (Healthcom). IEEE, 2018, pp. 1–6.
-